Electronic Colloquium on Computational Complexity, Report No. 189 (2013)

Overlays and Limited Memory Communication Mode(l)s

Periklis A. Papakonstantinou *
Tsinghua University

Dominik Scheder †
University of California, Berkeley

Hao Song ‡
Tsinghua University

December 21, 2013

Abstract

We give new characterizations and lower bounds relating classes in the communication complexity polynomial hierarchy and circuit complexity to limited memory communication models.

We introduce the notion of *rectangle overlay complexity* of a function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$. This is a natural combinatorial complexity measure in terms of combinatorial rectangles in the communication matrix of f. Furthermore, we consider memoryless and limited-memory communication models, originally introduced in [11] with slightly different terminology. In these communication models there are two parameters of interest: (i) the message length or *space*, and (ii) the number of memory *states*. Specifically, these are one-way protocols which proceed in rounds. In each round, Alice sends one message of bounded length to Bob; receiving a message from Alice, Bob has to decide on the spot whether to output 0 or 1, or to continue the protocol. If he decides to continue, he immediately forgets Alice's message. In memoryless protocols, no memory is transferred between different rounds (but Bob still has "space" to hold Alice's messages within each round). We can make Bob more powerful by giving him some constant size memory, which he can update at the end of each round.

We show that rectangle overlays completely characterize memoryless protocols. Then, we go on to show several connections to the communication complexity polynomial hierarchy defined by Babai, Frankl and Simon in 1986 [6]. This hierarchy has recently regained attention because its connection to the algebrization barrier in complexity theory [1]. We show that $\mathsf{P}^{\mathsf{NP}^{cc}}$ is completely characterized by memoryless protocols with $\mathsf{polylog}(n)$ space (message length), and thus it admits a purely combinatorial characterization in terms of rectangle overlays. If in addition Bob is given 3 states of memory besides $\mathsf{polylog}(n)$ space (message length), Alice and Bob can compute every level of Σ_k^{cc} in the communication complexity hierarchy (for constant k), and also every function in AC^0. Furthermore, we show that a 5-state Bob with $\mathsf{polylog}(n)$ space (message length) can compute exactly the functions in the communication class PSPACE^{cc}. This gives the first meaningful characterization of PSPACE^{cc} in terms of space, originally defined in [6] without any notion of space. We also study equivalences and separations between our limited memory communication model and branching programs, and relations to circuit classes.

*email: papakons@tsinghua.edu.cn
†email: dominik.scheder@gmail.com
‡email: hao.song42@gmail.com

ISSN 1433-8092

Contents

1 Introduction

Communication complexity is one of the jewels of theory of computation. There is a number of profound questions of interest to communication complexity itself, though its power mostly shows when proving lower bounds for other models of computation. For instance, lower bounds for streaming e.g. [4, 14]), circuit complexity e.g. [15, 20], and property testing [8].

In classical communication complexity, two players Alice and Bob are each given only part of the input to a computational problem. The goal is to cooperatively solve the problem by exchanging as little information as possible. The computational power of Alice and Bob is unlimited, in particular, they have unlimited time and unlimited memory space. In this two-player setting, we assume Alice and Bob receive inputs $x, y \in \{0,1\}^n$ respectively and wish to compute a boolean function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ on (x, y), that is to determine the value of $f(x, y)$.

In this work we revisit some novel one-way communication models, first introduced in [11] with slightly different terminology, [1] in which the memory of the receiving end is severely limited (in some cases, even without any explicit memory). These communication models find perhaps surprising connections to the communication classes corresponding to the polynomial time hierarchy [6]. These classes regained interest recently through their relation to algebrizing methods [1, 12]. Our models give rise to characterizations of different levels of this hierarchy, either through very simple and intuitive combinatorial characterizations or through our limited memory communication models. First, these characterization make transparent known separation results in the hierarchy (in particular those achieved recently by Impagliazzo and Williams [13]). In addition, we explore relationships between our models and bounded-width branching programs and bounded-depth circuit classes.

We hope that further important advancements in the aforementioned fields in complexity theory can be obtained through this rich set of connections. We discuss future research directions in Section 7.

1.1 Memoryless Communication Model and its Combinatorial Characterization

In the so-called *one-way memoryless* model, the protocols proceed in rounds. In round i, Alice sends a message $P_A(i, x)$ to Bob. Upon receiving this message, Bob has three choices: He can terminate and output 0; terminate and output 1; or decide to continue. If he continues, he forgets everything that has happened so far, and enters the next round. The *memoryless complexity* of the protocol is $\max_{x,i} |P_A(i, x)|$, i.e. the length of the longest message. The memoryless complexity of a function f, denoted by $S(f)$, is the complexity of the most efficient protocol that correctly computes f, i.e. $\min_{\mathcal{P}} \max_{x,i} |P_A(i, x)|$ where $\mathcal{P}$ ranges over all protocols that correctly compute f.

1.1.1 Rectangle Overlay

An advantage of this one-way memoryless model is that it has a very simple and intuitive combinatorial characterization: *rectangle overlays*. A rectangle overlay generalizes the rectangle partition

[1] We introduced new terminology since [11] reflecting our understanding on what matters. We promise that we won't change the terms again. In [11] Bob had 2 types of memory. A large oblivious one that was only a function of the communication, and a small non-oblivious one that was a function of everything. Now, in each round Bob receives a message of certain length (this corresponds to oblivious memory), and he maintains some memory states (this corresponds to the old definition of non-oblivious memory).

and rectangle cover concepts that have been previously used in the study of classical communication complexity. Similar to rectangle partitions and rectangle covers, rectangle overlays are defined using combinatorial rectangles of the form $R = X \times Y$, where $X, Y \subseteq \{0,1\}^n$. A *rectangle overlay of length l* is a sequence of tuples $(R_1, b_1), (R_2, b_2), \ldots, (R_l, b_l)$ where each R_i is a combinatorial rectangle that has "color" $b_i \in \{0,1\}$. For completeness, the union of these rectangles must cover the whole domain $\{0,1\}^n \times \{0,1\}^n$. A function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ is defined by this overlay in the following way: for each input pair $(x, y) \in \{0,1\}^n \times \{0,1\}^n$, find the first rectangle R_i in the sequence that contains (x, y), then define the output of $f(x, y)$ to be b_i. We denote minimum length of any rectangle overlay that computes f as $RO(f)$.

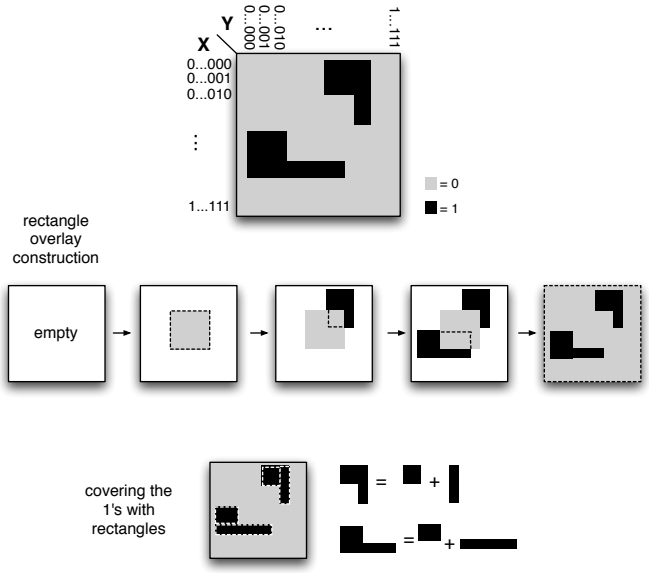

Figure 1: The top square is the communication matrix of function. Following that is a construction of this matrix by a rectangle overlay (we draw this using the "overlay" feature of image processing tools, hence the name). The third row shows a covering of the 1s by rectangles. We will see that rectangle overlays can be exponentially shorter in the number of steps than rectangle covers.

Our first result shows that the message length $S(f)$ used to compute f with one-way memoryless protocols is characterized by $RO(f)$, and it is tight up to a factor of 2.

Theorem 1. *For every boolean function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, it holds that*

$$S(f) \leq \lceil \log(RO(f)) \rceil \leq 2S(f) + 1$$

Let us demonstrate the rectangle overlay concept through the "Greater-Than" function GT: for $x, y \in \{0,1\}^n$, $\mathrm{GT}(x, y) = 1$ if and only if $x > y$ where $>$ is the lexicographic order. A standard fooling set argument (cf. [17] Section 1.3) shows that one needs at least $2^n - 1$ monochromatic rectangles to cover all the $x > y$ inputs; similarly for all the $x \leq y$ inputs. However, there exists a rectangle overlay of length $2n + 1$ that computes GT. Say $x = x_1 x_2 \ldots x_n$ and $y = y_1 y_2 \ldots y_n$. We

define $R_i^{>} \stackrel{\text{def}}{=} \{(x,y) \mid x_i = 1, y_i = 0\}$ and give it color 1; similarly, $R_i^{<} \stackrel{\text{def}}{=} \{(x,y) \mid x_i = 0, y_i = 1\}$ gets color 0. Finally, we set $R_{n+1} = \{0,1\}^n \times \{0,1\}^n$ and give it color 0. It remains to check that the rectangle overlay

$$(R_1^{>},1),(R_1^{<},0),(R_2^{>},1),(R_2^{<},0),\ldots,(R_n^{>},1),(R_n^{<},0),(R_{n+1},0)$$

computes GT. It does and it has length $2n+1$.

The optimal rectangle overlay for a function f is never longer than its optimal rectangle cover, and the GT example shows that the separation can be exponential. Using Theorem 1, we conclude that the memoryless complexity $S(f)$ of a function is never larger than its nondeterministic (or co-nondeterministic) communication complexity: $S(f) \leq \min(N^1(f), N^0(f))$. Furthermore, GT shows that $S(f)$ can be exponentially smaller than the (co-) nondeterministic communication complexity.

The rectangle overlay characterization allows a very intuitive lower bound technique for one-way memoryless protocols.

Theorem 2. *Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ be a boolean function and μ be a a product probability distribution μ on $\{0,1\}^n \times \{0,1\}^n$. Let $\epsilon_\mu \stackrel{def}{=} \max_R \mu(R)$, where R ranges over all monochromatic rectangles in the communication matrix of f. Then $S(f) \geq \frac{1}{4}\log\left(\frac{1}{\epsilon_\mu}\right) - \frac{1}{2}$.*

Using this lower bound technique we prove nearly tight lower bounds in the memoryless complexity of several functions, including the inner product function over $\mathsf{GF}(2)$ (Corollary 11), the "List-Non-Equality" function (Corollary 12) and the "Gap-Hamming-Distance" function (Corollary 13) [2].

Open Problem 1. *Suppose everything as in Theorem 2, and let $\epsilon \stackrel{def}{=} \min_\mu \epsilon_\mu$, where μ ranges over all product distributions. Is it true that $S(f) \leq \mathsf{poly}\big(\log\left(\frac{1}{\epsilon}\right)\big)$?*

1.1.2 The Class $\mathsf{P}^{\mathsf{NP}^{cc}}$

Babai, Frankl, and Simon [6] defined a communication analogue of the polynomial hierarchy of the Turing Machine world. This hierarchy is defined using non-determinism and alternating quantifiers, similarly to its Turing Machine counterpart. For completeness we recall the formal definitions of these communication classes in Section 2. For now, let us just say that the first level of this hierarchy $\Sigma_1^{cc} = \mathsf{NP}^{cc}$ is the class of all functions $f = \{f_n\}_{n=1}^{\infty}$ [3] whose nondeterministic communication complexity $N^1(f)$ is at most $\mathsf{polylog}(n)$. Similarly, $\Pi_1^{cc} = \mathsf{coNP}^{cc}$ contains those f's with $N^0(f) \in \mathsf{polylog}(n)$.

The class $\mathsf{P}^{\mathsf{NP}^{cc}}$ is defined using protocols with access to a certain *oracle*. Suppose Alice and Bob can exchange messages as usual, but in addition they can query *Oracles* in the class NP^{cc} (by each sending its part for the query). The cost for such a protocol is the total amount of communication, where an oracle query on strings of length s counts as $\lceil \log(s) \rceil$ bits of communication. Now, $\mathsf{P}^{\mathsf{NP}^{cc}}$ is the class of all functions computable by such an oracle protocol of cost at most $\mathsf{polylog}(n)$.

[2] Technically, the "Gap-Hamming-Distance" function is a partial function, we prove lower bound for some total-function extensions of this function

[3] Hereupon in this paper, whenever we discuss asymptotics we use the notation $f = \{f_n\}_{n=1}^{\infty}$ to denote a family of boolean functions, one for each input length n, $f_n : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$. The communication protocol (or branching program, or circuit) involved should also be understood as a non-uniform family of protocols (or branching programs, or circuits), one for each input length.

Theorem 3. *A function* $f = \{f_n\}_{n=1}^{\infty}$ *is in* $\mathsf{P}^{\mathsf{NP}^{cc}} \iff S(f) \in \mathsf{polylog}(n)$.

It's known that $\mathsf{P}^{\mathsf{NP}^{cc}}$ is strictly between the first and second level of the polynomial hierarchy [13].

Fact 4. $\Sigma_1^{cc} \cup \Pi_1^{cc} \subsetneq \mathsf{P}^{\mathsf{NP}^{cc}} \subsetneq \Sigma_2^{cc} \cap \Pi_2^{cc}$

Impagliazzo and Williams proved the second separation witnessed by the aforementioned "List-Non-Equality" function LNE: $S(\text{LNE}) = \Omega(\sqrt{n})$. Note that LNE can be computed by a depth-3 circuit of polynomial size. We improve this separation showing that there are two more functions of small depth-3 circuits but *linear* memoryless complexity: one of them in Σ_2^{cc} and the other in Π_2^{cc}. Both of these functions are total function extensions of the partial function "Gap-Hamming-Distance" (Corollary 13).

The complexity classes $\Sigma_k^{cc}, \Pi_k^{cc}$ have a nice combinatorial characterization as iterated intersections and unions of monochromatic rectangles [6]. Before our work no such characterization was known for $\mathsf{P}^{\mathsf{NP}^{cc}}$. We give a very elegant (and somewhat unexpected) combinatorial characterization for $\mathsf{P}^{\mathsf{NP}^{cc}}$, by first showing that $\mathsf{P}^{\mathsf{NP}^{cc}}$ is characterized by one-way memoryless protocols, and then showing the equivalence of our one-way memoryless model to rectangle overlays. In addition, we are now able to give very intuitive proofs for several results about $\mathsf{P}^{\mathsf{NP}^{cc}}$ in [13], including the above separation.

1.2 Limited Memory Communication Model

In the *one-way limited memory* communication model, we give Bob a constant size memory realized by constant many states. We stress out that this is a "permanent" memory that Bob has – not to be confused with the space occupied by the messages (kept temporarily for the duration of each round) Bob receives from Alice. That is to say, upon receiving the message from Alice, if Bob chooses to continue without giving a final answer, he can carry a constant amount of information to the next round. We henceforth denote the set of all functions computable by one-way limited-memory protocols where Bob uses an s-bit message receiving buffer and w memory states (or a roughly equivalent, $\lceil \log w \rceil$-bit memory) by $\mathsf{SPACE}_{\mathsf{LTD}}[s, w]$; the message is received in the beginning and the message buffer is emptied immediately after the communication round ends. A somewhat surprising fact is that, with 5 distinct memory states and $\mathsf{polylog}$ message length, we can exactly characterize the whole communication complexity polynomial hierarchy. (The class PSPACE^{cc} is defined to be the "infinite-level" limit of the hierarchy in [6].)

Theorem 5. $\mathsf{PSPACE}^{cc} = \mathsf{SPACE}_{\mathsf{LTD}}[\mathsf{polylog}(n), 5]$. *Moreover, the same holds true for every* ≥ 5 *constant many states.*

Despite what its name suggests, when the class PSPACE^{cc} was first defined by Babai, Frankl and Simon in 1986, the authors noted that "we do not have a notion corresponding to space". Now through this newly found equivalence, we can actually put a "polynomial" space notion behind this long known class PSPACE^{cc}. We emphasize that we really need this subtle notion of "space" in order to characterize PSPACE^{cc}, i.e. $\mathsf{polylog}(n)$ message length and these all-powerful 5 memory states (lacking those 5 states by Theorem 3 we go down to $\mathsf{P}^{\mathsf{NP}^{cc}}$).

If Bob has 3 memory states, and $\mathsf{polylog}$ message length, he can still compute any constant level of the communication polynomial hierarchy.

Theorem 6. *For every constant $k \in \mathbb{N}^+$ (independent of the input length n), $\Sigma_k^{cc} \cup \Pi_k^{cc} \subseteq \mathsf{SPACE}_{\mathsf{LTD}}[\mathsf{polylog}(n), 3]$.*

Now, we discuss connections of our models to circuit complexity classes. A corollary of the above theorem is that every function in AC^0 can be computed with 3 memory states and $\mathsf{polylog}(n)$ message length: $\mathsf{AC}^0 \subseteq \mathsf{SPACE}_{\mathsf{LTD}}[\mathsf{polylog}(n), 3]$. We say that an AC^0 function is contained in a communication class to denote that this holds under every equipartition (for Alice and Bob) of the input. Theorem 6 is proved by derandomizing Razborov and Smolensky's low-degree polynomial approximation of bounded-depth circuits [21, 23]. For this we borrow ideas from Newman's randomness reduction scheme [19] and Braverman's error indicator [10]. Allender and Hertrampf did something similar in their circuit depth reduction paper [3].

Furthermore, we note that Theorem 5 is the communication complexity analogue of Barrington's Theorem [7]. Barrington's Theorem states that every function in NC^1 can be computed by a width-5 branching program of polynomial length. In fact, the proof of Theorem 5 also shows the following.

Theorem 7. *Suppose $f \in \mathsf{NC}^1$. Then $f \in \mathsf{SPACE}_{\mathsf{LTD}}[O(\log n), 5]$ (under every input partition).*

Regarding branching programs we observe the following.

Fact 8. *For a boolean function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, if there is a width-$w$ branching program of length l that computes f, then $f \in \mathsf{SPACE}_{\mathsf{LTD}}[\lceil \log l \rceil + 1, w]$.*

The other direction is not true. Indeed, there is a natural function for which limited-memory protocols are more efficient than branching programs. Consider the inner product modulo 3: $\mathsf{IP}_3(x,y) \stackrel{\text{def}}{=} \sum_{i=1}^n x_i y_i \mod 3$. An unpublished result by Shearer [22] shows that every width-2 branching program computing IP_3 has length $\Omega\left(\left(\frac{3}{2\sqrt{2}}\right)^n\right)$. On the other hand, there is a protocol in which Bob has two memory states that computes IP_3 with $O(\sqrt{n})$ message length. More generally:

Theorem 9. *Suppose $f(x,y)$ can be decomposed as $g(h_1(x,y), \ldots, h_k(x,y))$ where each $h_i : \{0,1\}^n \to \{0,1\}$ depends on at most ℓ variables. Then $f \in \mathsf{SPACE}_{\mathsf{LTD}}[\ell + k + \lceil \log k \rceil, 2]$.*

The details of the proof of this theorem are deferred to Appendix B. We can decompose IP_3 into $\sqrt{n}$ blocks of $\sqrt{n}$ bits each. The value of IP_3 on one block is a number in $\{0,1,2\}$ and can be represented using two bits. Thus, $k = 2\lceil\sqrt{n}\rceil$ and $\ell = \lceil\sqrt{n}\rceil$, and $\mathsf{IP}_3 \in \mathsf{SPACE}_{\mathsf{LTD}}[O(\sqrt{n}), 2]$.

1.3 Roadmap to the Rest of the Paper

We recall basic definitions in Section 2. Formal definitions of our communication model are given in Section 3. Section 3 also contains our results concerning rectangle overlays and some preliminary discussions and relations to circuit classes (in the Appendix we give an example of a function separating our memoryless model from AC^0). Sections 4, 5, 6 contain characterizations and separations of $\mathsf{P}^{\mathsf{NP}^{cc}}$, Σ_k^{cc}, and PSPACE^{cc}. In Section 7 we discuss some future research directions.

2 Standard models: definitions and notation

This paper studies relations of communication memory models with boolean circuits and branching programs. For a detailed treatment cf. [5, 25]. For completeness we briefly recall the definitions

here. We also recall the definitions of oracle classes in Communication Complexity [6]. For a comprehensive treatment and definitions of deterministic and non-deterministic communication complexity cf. [17] (Section 1.1 and 2.1).

Definition 1 (Boolean Circuits)**.** *A* boolean circuit *is a directed acyclic graph (DAG) with a unique sink. Each source of this graph corresponds to a boolean input; each internal node has a label from the set* $\{\wedge, \vee, \neg\}$ *denoting the boolean operation (gate) to apply (the in-degree of* $\neg$ *nodes must be* 1*). The* fan-in *of a node is its in-degree, and the* fan-out *of a node is its out-degree. We evaluate the output of the circuit inductively by evaluating the corresponding gates. The* depth *of a circuit is the length of the longest path from one of the sources to the unique sink.*

Definition 2 (Bounded-Depth Circuit Classes: AC^0 and NC^1)**.** *A family* $\{f_n\}_{n=1}^{\infty}$ *of boolean functions is in* AC^0 *if every* f_n *can be computed by a boolean circuit on the basis* $\{\vee, \wedge, \neg\}$ *with constant depth, polynomial size and unbounded fan-in. A family* $\{f_n\}_{n=1}^{\infty}$ *is in* NC^1 *if every* f_n *can be computed by a boolean circuit on the basis* $\{\vee, \wedge, \neg\}$ *with* $O(\log n)$ *depth, polynomial size and fan-in* 2*.*

Definition 3 (Branching Programs)**.** *Let* $w, l, n \in \mathbb{N}^+$*. A width* w *length* l *branching program defined on every a* n*-bit input, is defined as a sequence of* l *instructions and a set* $S_{\text{YES}} \subseteq \{1, 2, \ldots, w\}$*. Each instruction is of the form* $< i, f, g >$*, where* $i \in \{1, 2, \ldots, n\}$ *and* f, g *are functions from* $\{1, 2, \ldots, w\}$ *to* $\{1, 2, \ldots, w\}$*. Such a branching program computes a function* $B : \{0,1\}^n \to \{0,1\}$ *with the computation defined as follows: A* state M *is initially set to* 1*. We execute the instructions in the branching program in order: For an instruction* $< i, f, g >$*, we read the* i^{th} *bit of the input. If this is* 0*, update* $M \stackrel{def}{=} f(M)$*; if it is* 0*, update* $M \stackrel{def}{=} g(M)$*. If the state* M *is in* S_{YES} *after the final instruction, output* 1*. Otherwise output* 0*.*

Now, we recall the related communication complexity classes: the polynomial hierarchy, PSPACE^{cc}, and oracle communication classes.

Definition 4 (Communication Complexity Polynomial Hierarchy)**.** *For* $k(n) \leq \mathsf{polylog}(n)$*. We say that a family of boolean functions* $\{f_n\}_{n=1}^{\infty}$ *can be computed by a* $\Sigma_{k(n)}^{cc}$ *protocol, if there exist* $\phi, \psi : \{0,1\}^n \times \{0,1\}^* \to \{0,1\}$*, such that for every two* n*-bit strings* x, y*,* $f_n(x, y) = 1$ *if and only if*

$$\exists u_1 \forall u_2 \exists u_3 \ldots Q_{k(n)} u_{k(n)} (\phi(x, u) \diamond \psi(y, u))$$

Here $u \stackrel{def}{=} u_1 u_2 \ldots u_{k(n)}$ *is a bit string of length at most* $\mathsf{polylog}(n)$*. When* k *is even,* $\diamond$ *stands for* $\vee$ *and* $Q_{k(n)}$ *is* $\forall$*; when* k *is odd,* $\diamond$ *stands for* $\wedge$ *and* $Q_{k(n)}$ *is* $\exists$*.*

We define $\Pi_{k(n)}^{cc}$ *analogously, by switching the roles of the two quantifiers (*$\exists$ *and* $\forall$*) and the two boolean operators (*$\wedge$ *and* $\vee$*).*

Furthermore, we define $\mathsf{PSPACE}^{cc} \stackrel{def}{=} \Sigma_\infty = \Pi_\infty$ *(although here by definition* ∞ *is restricted to* $\mathsf{polylog}(n)$*).*

For example, one checks that Σ_1^{cc} is the set of all families $\{f_n\}_{n=1}^{\infty}$ that have $\mathsf{polylog}(n)$ nondeterministic communication complexity.

We remark that unlike the Turing Machine complexity world the above definition uses two predicates (ϕ and ψ) each of which is individually given Alice's (x) and Bob's (y) input.

Observation 10 (Relations between Communication Classes and Circuit Classes)**.** *Let* $\{f_n\}_{n=1}^{\infty}$ *be a family of functions* $f_n : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$. $\{f_n\}_{n=1}^{\infty} \in \Sigma_{k(n)}^{cc}$ *if and only if there exist* $m \leq 2^{\mathsf{polylog}(n)}$*, functions* $A, B : \{0,1\}^n \to \{0,1\}^m$*, and a circuit* C *on* $2m$ *input variables such that:*

- *The depth of* C *is at most* $k(n)+1$.
- *The output gate of* C *is* $\vee$.
- C *has size at most* $2^{\mathsf{polylog}(n)}$.
- $f(x,y) = C(A(x), B(y))$.

Similarly, $\{f_n\}_{n=1}^{\infty} \in \Pi_{k(n)}^{cc}$ *if and only if the circuit is as described, but with a* $\wedge$*-gate at the top.*

Definition 5 ($\mathsf{P}^{\mathsf{NP}^{cc}}$)**.** *An* oracle protocol *is defined just like classical communication protocols, where instead of only communicating between each other, Alice and Bob can also make queries of the form* $q(f(x), g(y))$*. Here* f *and* g *are arbitrary preprocessing functions with matching output length* m*, and* q *is a query function from a class specified by the model. The total cost of the protocol is communication cost plus query cost. Each communication bit has cost* 1*. Each oracle query, with output length* m *preprocessing functions, has cost* $\log m$.

The class $\mathsf{P}^{\mathsf{NP}^{cc}}$ *contains exactly those protocols with total cost* $\mathsf{polylog}(n)$ *when allowed to query functions in* NP^{cc}.

3 Model Definition and a Combinatorial Characterization

Here we provide a formal definition of our one-way limited memory communication model, related complexity measures, complexity classes and the rectangle overlay characterization. Also, at the end of this section we list some important consequences of these definitions and characterization.

Definition 6. *In a* one-way limited-memory *protocol there are two parameters of interest: message length* s *and number of memory states* w*. In particular, every message received by Bob will be of length* $\leq s$ *(he receives this in an atomic form, i.e. it can be thought as if this message were stored temporarily in a buffer occupying some space for the duration of the round). We denote the content of this message by a variable* $\alpha \in \{0,1\}^s$*. In addition, Bob has some limited memory states, where the current state is represented by a variable* $\gamma \in \Gamma$*, here* Γ *is the set of memory states. (*$|\Gamma| = w$*, throughout this paper; we will focus on the case where* w *is a small constant integer, typically* ≤ 5*).*

The protocol proceeds in rounds. In the i*-th round (*$i = 1, 2, \ldots$*), Alice uses an* upload function $P_A : \mathbb{N}^+ \times \{0,1\}^n \to \{0,1\}^s$ *to compute an* s*-bit long message* $P_A(i,x)$ *to be placed into Bob's side. Upon receiving a message* α *from Alice, Bob uses his state transition function* $T_B : \{0,1\}^n \times \Gamma \times \{0,1\}^s \to \{0,1,\perp\} \times \Gamma$ *to compute* $T_B(y,\gamma,\alpha)$*. Suppose the output is* (β,γ')*. If* $\beta = 0$ *or* $\beta = 1$*, Bob outputs* β *as the final answer of the protocol and halts the protocol; otherwise (*$\beta = \perp$*) Bob continues to the next round with the new memory state* γ'*.* [4]

[4]Note that both Alice and Bob are all-powerful when doing local computations, that is to say, both P_A and T_B are arbitrary functions.

The protocol must halt within $w \cdot 2^s$ many rounds. We call a protocol memoryless *whenever $w = 1$.* [5]

Definition 7. *$\mathsf{SPACE}_{\mathsf{LTD}}[s, w]$ is the set of all functions computable by one-way limited-memory protocols where Bob receives messages of length $\leq s$ and a memory of size w (i.e. the number of states). We write $\mathsf{SPACE}_{\mathsf{LESS}}[s] \stackrel{def}{=} \mathsf{SPACE}_{\mathsf{LTD}}[s, 1]$.*

3.1 The Rectangle Overlay Characterization

As already discussed in the introduction one of the main contributions of this paper is a combinatorial characterization of one-way memoryless protocols through *rectangle overlays*. Below is the formal definition and the proofs of the characterization theorems.

Definition 8. *For a positive integer n, a* rectangle overlay on $\{0,1\}^n \times \{0,1\}^n$ of length l *is a sequence of tuples $(R_1, b_1), (R_2, b_2), \ldots, (R_l, b_l)$ such that*

- *for each $i \in \{1, 2, \ldots, l\}$, R_i is a combinatorial rectangle on the domain $\{0,1\}^n \times \{0,1\}^n$ (in other words, $R_i = X_i \times Y_i$ where $X_i \subseteq \{0,1\}^n$ and $Y_i \subseteq \{0,1\}^n$) and $b_i \in \{0,1\}$*
- *$\bigcup_{i=1}^{l} R_i = \{0,1\}^n \times \{0,1\}^n$*

Such a rectangle overlay computes/defines a function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ in a natural way. For every input pair $(x, y) \in \{0,1\}^n \times \{0,1\}^n$, find the smallest i such that $(x, y) \in R_i$, then define the output of $f(x, y)$ to be b_i.

The rectangle overlay complexity *of a function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, denoted as $RO(f)$, is defined to be $RO(f) \stackrel{def}{=} \min\{l$: there is a rectangle overlay of length l that defines $f\}$.*

Now, we are ready to give the proof of Theorem 1 (p. 2).

Proof of Theorem 1. First we show that $S(f) \leq \lceil \log(RO(f)) \rceil$. Let $m = \lceil \log(RO(f)) \rceil$. There is a rectangle overlay $(R_1, b_1), (R_2, b_2), \ldots, (R_l, b_l)$ that computes f, and $l \leq 2^m$. We construct a one-way memoryless protocol with message length requirement at most m. For each $1 \leq i \leq l$ write $R_i = X_i \times Y_i$, where $X_i, Y_i \subseteq \{0,1\}^n$. Given input $(x, y) \in \{0,1\}^n \times \{0,1\}^n$, let $\{X_{i_1}, X_{i_2}, \ldots, X_{i_{l'}}\}$ is the set of all X_i's such that $x \in X_i$, and $i_1 < i_2 < \ldots < i_{l'}$. Our protocol works as follows: In round j, Alice's upload function is $P_A(j, x) \stackrel{\text{def}}{=} i_j$. Bob's state transition function $T_B(y, i)$ outputs b_i if $y \in Y_i$, or $\perp$ if otherwise. Clearly, $l' \leq l \leq 2^m$, therefore the message length requirement of this protocol is at most m.

Second, we show that $\lceil \log(RO(f)) \rceil \leq 2S(f)+1$. Let $s \stackrel{\text{def}}{=} S(f)$. There is a one-way memoryless protocol with message length s that computes f. We construct a rectangle overlay of length at most 2^{2s+1} that computes f. According to Definition 6, there are functions T_B and P_A such that in the i-th step, Bob uses $T_B(y, P_A(i, x))$ to determine whether to give the final answer from the set $\{0,1\}$ or continue with the computation. For each tuple $(i, \alpha, b) \in \{1, 2, \ldots, 2^s\} \times \{0,1\}^s \times \{0,1\}$, we define $X_{i,\alpha,b} \stackrel{\text{def}}{=} \{x \mid P_A(i, x) = \alpha\}$, $Y_{i,\alpha,b} \stackrel{\text{def}}{=} \{y \mid T_B(y, \alpha) = b\}$ and $R_{i,\alpha,b} \stackrel{\text{def}}{=} X_{i,\alpha,b} \times Y_{i,\alpha,b}$.

[5] It's easy to see that in the *memoryless* model, where the sole memory state in Γ of Bob is useless, if a problem can be computed within s message length, the optimal protocol will always halt within 2^s rounds, so the round number limit doesn't really matter. But in a model with at least 2 memory states, the model would be kind of uninteresting without the round number limit, because every problem would be computable within $O(\log n)$ message length without the round number limit.

Note that for fixed i, the rectangles $R_{i,\alpha,b}$ are disjoint. We order the rectangles $\{R_{i,\alpha,b}\}_{(i,\alpha,b)}$ in the increasing order of i and color each rectangle $R_{i,\alpha,b}$ with color b. This gives a rectangle overlay that computes f and has length 2^{2s+1}. □

Proof of Theorem 2. Let $\ell = RO(f)$. There is a rectangle overlay (R_1, b_1), (R_2, b_2), …, (R_ℓ, b_ℓ) that computes f. Then according to Theorem 1, $S(f) \geq \frac{\log(\ell)}{2} - \frac{1}{2}$.

Next we will construct a sequence of rectangles $T_0, T_1, T_2, \ldots, T_\ell$ such that: (i) $T_0 = \{0,1\}^n \times \{0,1\}^n$; (ii) $T_\ell = \emptyset$; (iii) $R_j \cap T_i = \emptyset$ for all $1 \leq j \leq i \leq \ell$; (iv) for every $i \in \{1, 2, \ldots, \ell\}$, we have $T_i \subseteq T_{i-1}$, and (v) $\mu(T_i) \geq \mu(T_{i-1}) - \sqrt{\epsilon}$. The existence of a sequence implies that $\ell \geq \frac{1}{\sqrt{\epsilon}}$ and $S(f) \geq \frac{\log(\ell)}{2} - \frac{1}{2} \geq \frac{1}{4}\log\left(\frac{1}{\epsilon}\right) - \frac{1}{2}$.

We construct the sequence $\{T_i\}$ as follows: Set $T_0 = \{0,1\}^n \times \{0,1\}^n$. We define $T_1, T_2, \ldots, T_\ell$ inductively. For every $i \in \{1, 2, \ldots, \ell\}$, define $\bar{R}_i \stackrel{\text{def}}{=} R_i \cap T_{i-1}$. $\bar{R}_i$ is clearly a combinatorial rectangle, and by property (iii) it is disjoint from all previous $R_1, \ldots, R_{i-1}$. Thus, the function defined by the rectangle overlay outputs color b_i for all $(x, y) \in \bar{R}_i$. In other words, $\bar{R}_i$ is monochromatic. Write $T_{i-1} = X_{i-1} \times Y_{i-1}$ and $\bar{R}_i = \bar{X}_i \times \bar{Y}_i$. Since μ is a product distribution, we get $\mu(\bar{R}_i) = \mu(\bar{X}_i) \cdot \mu(\bar{Y}_i)$ and $\min\{\mu(\bar{X}_i), \mu(\bar{Y}_i)\} \leq \sqrt{\mu(\bar{R}_i)} \leq \sqrt{\epsilon}$. Then we define

$$T_i \stackrel{\text{def}}{=} \begin{cases} (X_{i-1} \setminus \bar{X}_i) \times Y_{i-1} & \text{if } \mu(\bar{X}_i) \leq \mu(\bar{Y}_i) \\ X_{i-i} \times (Y_{i-1} \setminus \bar{Y}_i) & \text{if } \mu(\bar{X}_i) > \mu(\bar{Y}_i) \end{cases}$$

In words, we obtain T_i by cutting a piece away from T_{i-1} to ensure that T_i and R_i are disjoint. Thus, T_i satisfies (iii) and (iv). Finally, the piece we cut away has weight at most $\sqrt{\epsilon}$, thus (v) is satisfied, too.

Since $\cup_{i=1}^{2^m} R_i = \{0,1\}^n \times \{0,1\}^n$ it holds that $T_\ell = \emptyset$. This completes the construction of the sequence and concludes the proof. □

3.2 Consequences of the Rectangle Overlay Characterization

The above characterization significantly simplifies and strengthens lower bounds proved previously (as in e.g. [11]), and significantly simplifies and appropriately conceptualizes [13]. Also, involving the isoperimetric properties of the Hamming Cube we obtain Lemma 13 whose proof is given in the Appendix.

Corollary 11 (Theorem 22 in [11])**.** $S(\mathrm{IP}) \geq \frac{n}{4} - \frac{1}{2}$. *Here* IP *is the Inner-Product function for two n-dimensional verctors over* $\mathsf{GF}(2)$.

Proof. It is well-known that every monochromatic rectangle of Inner Product has size at most 2^n. See for example Example 1.25 in the book by Kushilevitz and Nisan [17]. Let μ be the uniform distribution on $\{0,1\}^n \times \{0,1\}^n$. Thus $\mu(R) \leq 2^{-n}$ for every monochromatic rectangle R in communication matrix. □

Corollary 12. $S(\mathrm{LNE}_{\sqrt{n},\sqrt{n}}) \geq \frac{1}{2}\sqrt{n} - \frac{1}{4}\log n - \frac{1}{2}$. *Here* $\mathrm{LNE}_{b,k}$ *is a boolean function over two (bk)-bit inputs $x = x_{1,1}x_{1,2}\ldots x_{1,k}x_{2,1}\ldots x_{b,k}$ and $y = y_{1,1}y_{1,2}\ldots y_{1,k}y_{2,1}\ldots y_{b,k}$ (that is, each input is divided into b blocks, each block is k-bit long, e.g. $x_{3,5}$ is the fifth bit in the third block). And* $\mathrm{LNE}_{b,k}(x, y)$ *outputs* 1 *if and only if for every block i, there is at least one bit position j within this block such that $x_{i,j} \neq y_{i,j}$.*

Proof. Choose μ to be the uniform distribution on $\{0,1\}^n \times \{0,1\}^n$. In the communication matrix of $\mathrm{LNE}_{b,k}$, every 0-monochromatic rectangle R_0 satisfies $\mu(R_0) \leq b^2 \cdot 2^{-2k}$, and every 1-monochromatic rectangle R_1 satisfies $\mu(R_1) \leq 2^{-2b}$. See Impagliazzo and Williams [13] for a proof. We simply apply Theorem 2, substitute in $b = k = \sqrt{n}$, and the proof is done. □

Note that this lower bound, together with the fact that $\mathrm{LNE}_{\sqrt{n},\sqrt{n}} \in \Sigma_2^{cc} \cap \Pi_2^{cc}$ [18], gives a separation $\mathsf{P}^{\mathsf{NP}^{cc}} \subsetneq \Sigma_2^{cc} \cap \Pi_2^{cc}$.

Lemma 13. *There exists $f = \{f_n\}_{n=1}^{\infty} \in \Sigma_2^{cc}$, and $g = \{g_n\}_{n=1}^{\infty} \in \Pi_2^{cc}$, both extensions of the partial function "Gap Hamming Distance" GHD, such that $S(f) = \Omega(n)$ and $S(g) = \Omega(n)$.*

Here the partial function "Gap Hamming Distance", denoted as GHD, is defined as: for two n-bit strings x and y, if the Hamming distance $d_H(x,y) \leq n/3$, then $\mathrm{GHD}(x,y) = 0$; if $d_H(x,y) \geq 2n/3$, then $\mathrm{GHD}(x,y) = 1$; in between, GHD is not defined.

Proof of this lemma is given in Appendix A. An important consequence of the proof of this lemma is that f and g are both computable by depth 3 AC^0 circuits, therefore it is not the case that rectangle overlays can characterize (promise) AC^0. In particular, this shows that the intuition that AC^0 can be characterized by short rectangle overlays is false.

4 $\mathsf{P}^{\mathsf{NP}^{cc}}$ and Memoryless Protocols

Theorem 3 talks about the equivalence between $\mathsf{P}^{\mathsf{NP}^{cc}}$ and memoryless protocols of polylogarithmic message length.

Proof of Theorem 3. We prove that $\mathsf{SPACE}_{\mathsf{LESS}}[\mathsf{polylog}(n)] = \mathsf{P}^{\mathsf{NP}^{cc}}$. Let us start with the easier inclusion: $\mathsf{SPACE}_{\mathsf{LESS}}[\mathsf{polylog}(n)] \subseteq \mathsf{P}^{\mathsf{NP}^{cc}}$. The query function we will use here is the NP^{cc} "complete" function [6] "intersect", denoted by INT. For two n-bit strings $x = x_1 x_2 \ldots x_n$ and $y = y_1 y_2 \ldots y_n$, $\mathrm{INT}(x,y) = 1$ if and only if there exists $i \in \{1,2,\ldots,n\}$ such that $x_i = y_i = 1$.

Suppose $f = \{f_n\}_{n=1}^{\infty}$ is in $\mathsf{SPACE}_{\mathsf{LESS}}[\mathsf{polylog}(n)]$. By Theorem 1, each f_n has a rectangle overlay of length $\ell(n) \leq 2^{\mathsf{polylog}(n)}$. Let us denote this rectangle overlay by $(R_1, b_1), (R_2, b_2), \ldots, (R_{\ell(n)}, b_{\ell(n)})$. Write $R_i = X_i \times Y_i$ for each $i \in \{1,2,\ldots,\ell(n)\}$.

Alice and Bob preprocess x and y into two $\ell(n)$-bit strings $\hat{x}, \hat{y} \in \{0,1\}^{\ell(n)}$ as follows: For each $1 \leq i \leq \ell(n)$, set $\hat{x}_i$ to be 1 if $x \in X_i$ and 0 otherwise; define $\hat{y}$ analogously. Given these two $\ell(n)$-bit strings, Alice and Bob plan to find the smallest i such that $\hat{x}_i = \hat{y}_i = 1$ and output b_i. They can find this i using binary search, by quering INT at most $\mathsf{polylog}(n)$ many times, each query of length at most $2^{\mathsf{polylog}(n)}$. This gives us an oracle protocol in $\mathsf{P}^{\mathsf{NP}^{cc}}$ and shows that $f \in \mathsf{P}^{\mathsf{NP}^{cc}}$.

Second, we prove $\mathsf{P}^{\mathsf{NP}^{cc}} \subseteq \mathsf{SPACE}_{\mathsf{LESS}}[\mathsf{polylog}(n)]$. Let $f = \{f_n\}_{n=1}^{\infty}$ be in $\mathsf{P}^{\mathsf{NP}^{cc}}$. Consider f_n, let $\mathcal{P}$ be the corresponding oracle protocol and $\mathcal{T}$ its protocol tree. According to Definition 5, $\mathcal{T}$ has maximum depth $\mathsf{polylog}(n)$, and every query node makes a query to a NP^{cc} function of input length $2^{\mathsf{polylog}(n)}$.

We first observe that by definition, for every NP^{cc} function Q, there exists a family of combinatorial rectangles $\{R_i\}$, such that for each (x,y), $Q(x,y) = 1$ if and only there is i such that $(x,y) \in R_i$, and this i is a *certificate* for (x,y). Each certificate is of length $\mathsf{polylog}$ to the input

[6] INT is complete for NP^{cc} under the so-called "rectangle reduction", please refer to [6] for more details.

length of Q. In particular, since each query in $\mathcal{T}$ has input length $2^{\mathsf{polylog}(n)}$, each certificate for each of these queries is still of length $\mathsf{polylog}(n)$.

For each input (x, y) to f_n, we can describe the computational history the protocol will follow in $\mathcal{T}$ as follows: first we use a $\mathsf{polylog}(n)$-bit string p to denote all the communication bits and query answers along the way, from the root down to one of the leaf nodes; then we scan the path from root the leaf, for each query that answers 1, we concatenate one of the certificates for (x, y). This gives a string $(p, c_1, c_2, \ldots, c_t)$, in which $p, c_1, c_2, \ldots, c_t$ are all of $\mathsf{polylog}(n)$ length and $t = O(\mathsf{polylog}(n))$. Therefore the whole string is of length $\mathsf{polylog}(n)$.

Now let H is the set of all possible computational histories for all possible input pairs (x, y). We construct a one-way oblivious protocol $\mathcal{P}'$ for f in the following way: Alice enumerates through all the computational histories in H that are compatible to her input x one-by-one, lexicographically decreasing in p. That is, if $h = (p, c_1, \ldots, c_t)$ and $h' = (p', c'_1, \ldots, c'_{t'})$ are two histories with $p < p'$ (in lexicographic order), then Alice enumerates h' before h.

Each time Alice uploads the computational history to Bob's oblivious memory, Bob checks to see if this computational history is also compatible to his input y. If so, he outputs the label of the corresponding leaf node, otherwise he just continues.

Clearly $\mathcal{P}' \in \mathsf{SPACE}_{\mathsf{LESS}}[\mathsf{polylog}(n)]$. Let us prove that $\mathcal{P}'$ correctly computes f. Consider an input pair (x, y). Let p be root-to-leaf path in $\mathcal{T}$ followed by the original protocol $\mathcal{P}$ on input (x, y), and let h be the corresponding history as defined above. Let $h^* = (p^*, c_1^*, \ldots, c_t^*)$ be the computational history accepted by Alice and Bob in the protocol $\mathcal{P}'$. We will prove that $p = p^*$ by contradiction. Suppose $p \neq p^*$. Let v be the last node in $\mathcal{T}$ that is common to p and p^*.

- v cannot be an Alice's node or a Bob's node: In that case, one of Alice and Bob would reject p^* as incompatible to their input.
- So v must be a query node.
  - Suppose p takes the 0-child of v whilst p^* takes the 1-child of v. This is impossible: The correct answer for the query at node v is 0 (since p is the correct path), so either Alice or Bob would have rejected the purported 1-certificate in h^*.
  - Suppose p takes the 1-child of v whilst p^* takes the 0-child of v. Then $p > p^*$ lexicographically, and thus Alice enumerates h before h^*. Since h, being the correct history, is consistent for both Alice and Bob, they would have accepted it.

Thus $p = p^*$ and $\mathcal{P}'$ is correct. We conclude that $f \in \mathsf{SPACE}_{\mathsf{LESS}}[\mathsf{polylog}(n)]$. □

5 Constant levels of Σ_k^{cc} and 3-state protocols

In this section, for convenience, we define $\mathsf{PH}^{cc} \stackrel{\text{def}}{=} \bigcup_{k=1}^{\infty} \Sigma_k^{cc}$. Here k ranges over all constant (independent of input length n) positive integers. We prove Theorem 6: $\mathsf{PH}^{cc} \subseteq \mathsf{SPACE}_{\mathsf{LTD}}[\mathsf{polylog}(n), 3]$.

Proof outline: First, we show that for an arbitrary function $f \in \mathsf{PH}^{cc}$, there is a *randomized* protocol that computes f, in which Bob has only two states. That is, on every $(x, y) \in \{0,1\}^n \times \{0,1\}^n$, the protocol outputs the correct value with probability at least $2/3$. Second, we show how to make this protocol zero-error. Finally, we show that every zero-error protocol can be completely derandomized by giving Bob one additional memory state.

5.1 Bounded-error and zero-error protocols

In a bounded-error protocol, Alice and Bob share a source of randomness: At the beginning of the protocol, Alice and Bob are given the same random infinitely long bit string $r_1 r_2 r_3 \ldots$, and Alice's messages can depend on r: $P_A(i, x, r)$ instead of $P_A(i, x)$. Similarly, Bob can base his decision to output 0, output 1, or continue, on this random string. A protocol is a *bounded-error protocol computing* $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ if for each $(x, y) \in \{0,1\}^n \times \{0,1\}^n$, it outputs the correct value $f(x, y)$ with probability at least $2/3$. In the proof of Theorem 14 that follows we describe a zero-error protocol (see below for a definition), where in the first few steps of the construction we give a bounded-error one.

As in bounded-error protocols, in zero-error protocols Alice and Bob share a random string r. In a zero-error protocol, Bob has the additional possibility to abort the protocol and output "don't know". A protocol is a *zero-error protocol computing* $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ if for each $(x, y) \in \{0,1\}^n \times \{0,1\}^n$, (i) the output of the protocol is either $f(x, y)$ or "don't know", and (ii) Bob outputs "don't know" with probability at most $1/2$.

Theorem 14. *Let $f \in \mathsf{PH}^{cc}$. There exists a zero-error protocol computing f where Bob uses $\mathsf{polylog}(n)$ message length and has two memory states.*

Actually, we do not know whether every bounded-error protocol can be made zero-error without increasing the message length or Bob's memory. In the special case of $f \in \mathsf{PH}^{cc}$, it turns out that we can: The trick is to use not only a Razborov-Smolensky approximation for f, but also an "error-indicator", in spirit similar to Braverman [10].

Proof of Theorem 14. Let $f \in \mathsf{PH}^{cc}$. That is, $f \in \Sigma_k^{cc}$ for some k (that is independent on the input length n). By the connection between complexity classes and circuits (Observation 10), there is an $m \in 2^{\mathsf{polylog}(n)}$, functions $A, B : \{0,1\}^n \to \{0,1\}^m$, and an AC^0-circuit C over variables $u, v \in \{0,1\}^m$ such that $f(x, y) = C(A(x), B(y))$. We approximate C using the well-known method of Razborov and Smolensky [21, 23].

Lemma 15 (Razborov and Smolensky [21, 23])**.** *Given a circuit C of depth d and size S on the basis $\{\wedge, \vee, \oplus, \neg\}$ with unbounded fan-in on an m-bit input x, there is a scheme of assigning random polynomials over $\mathsf{GF}(2)$ to each gate in the circuit (the additive identity 0 corresponds to boolean value false, and the multiplicative identity 1 corresponds to boolean value true), such that*

- *For each gate g in $\mathcal{C}$, denote the random polynomial chosen as p_g. The degree of p_g is at most $(\log(3S))^d$.*
- *For every n-bit input x, we have*

$$Pr[\exists \text{ gate } g \in \mathcal{C} \text{ such that } g(x) \neq p_g(x)] \leq \frac{1}{3} .$$

  Here, $g(x)$ is the output of gate g.
- *If g is an input gate with input x_i, the i-th bit in x, then $p_g = x_i$.*
- *If g is a $\neg$-gate and c is its only child, then $p_g = p_c + 1$*
- *If g is an $\oplus$-gate with children $\{c_1, c_2, \ldots, c_t\}$, then have $p_g = \sum_{i=1}^{t} p_{c_i}$*

- *If g is a $\vee$-gate with children $\{c_1, c_2, \ldots, c_t\}$ and $c_1(x) = \cdots = c_t(x) = 0$, then $g(x) = 0$, too.*

Note that the lemma is already enough to obtain a bounded-error protocol evaluating C: Use common randomness to approximate the output gate by a $\mathsf{GF}(2)$-polynomial of polylogarithmic degree. Alice and Bob can evaluate this polynomial, provided Bob has space for messages of length $\mathsf{polylog}(n)$, and Bob has at least two memory states. They also can evaluate f, by precomputing $u = A(x)$ and $v = B(y)$ and then evaluating the $\mathsf{GF}(2)$-polynomial with u and v as input.

We show how to make this protocol zero-error. By applying DeMorgan's rule, We can assume our circuit C consists only of $\vee$-, $\oplus$-, and $\neg$-gates. The last four bullet points of Lemma 15 imply the following: If $g(x) = p_g(x)$ for all $\vee$-gates $g \in C$, then indeed $g(x) = p_g(x)$ for all gates. In other words, errors can only be introduced at $\vee$-gates. Such an error is introduced at $\vee$-gate g if and only if $g(x) = 0$ and $c_1(x) \vee \cdots \vee c_t(x) = 1$, where $c_1, \ldots, c_t$ are the children of g. That is, if and only if

$$\mathcal{E}_g(x) \stackrel{\text{def}}{=} \bigvee_{i=1}^{t} (\neg g(x)) \wedge c_i(x) \ .$$

evaluates to 1. Note that each disjunct in this expression is itself a polynomial of polylogarithmic degree. We conclude: If there is a gate g in C such that $g(x) \neq p_g(x)$, then

$$\mathcal{E}(x) \stackrel{\text{def}}{=} \bigvee_{\vee-\text{gates } g} \bigvee_{i=1}^{t} (\neg g(x)) \wedge c_i(x)$$

evaluates to 1.

We are ready to state our zero-error protocol for evaluating the circuit C. Alice and Bob evaluate $\mathcal{E}(x)$ by iteratign over all $\vee$-gates g and all children c_i of g. They evaluate the $\mathsf{GF}(2)$-polynomial $(\neg g(x)) \wedge c_i(x)$ with message length $\mathsf{polylog}(S)$ and two memory states. If this polynomial evaluates to 1, Bob immediately outputs "don't know" and stops. If they iterate over all $\vee$-gates g and c_i without Bob outputting "don't know", they know that $\mathcal{E}(x) = 0$. This implies that $g(x) = p_g(x)$ for all gates g in the circuit, including the output gate. Thus, they evaluate $p_g(x)$ for the output gate g and output its value.

What is the probability that Bob answers "don't know"? This can only happen if some error is introduced at a $\vee$-gate. This happens with probability at most $1/3$. Finally, Alice and Bob evaluate f by evaluating C on $A(x)$ and $B(y)$. This establishes a zero-error protocol for $f \in \mathsf{PH}^{cc}$ and completes the proof of Theorem 14. □

5.2 Derandomization of Zero-Error Protocols

Theorem 16 (Derandomization of Zero-Error Protocols)**.** *Suppose there is a zero-error protocol computing f with message length s and in which Bob has w memory states. Then there is a deterministic protocol computing f with $s + \lceil \log(3n) \rceil$ message length in which Bob has $w + 1$ memory states.*

The idea behind this proof is that Alice and Bob can iterate over all possible random strings r and simulate the zero-error protocol with this random string. If the simulated zero-error protocol causes the simulated Bob to output "don't know", Bob can remember this by moving into his

$w+1^{\text{st}}$ state. Another challenge is that Alice and Bob can afford to iterate over all random strings only if the number of all random strings is small. Here, we use a technique similar to the proof of Newman's Theorem that replaces public by private randomness (See for example Theorem 3.14 of [17]). We now prove Theorem 14 and Theorem 16.

Proof of Theorem 16. Suppose Alice and Bob compute f using a zero-error protocol with s message length and in which Bob has w memory states.

Denote by $P(x,y,r)$ the output of that protocol on input (x,y), when the common random string is r. Clearly, $P(x,y,r) \in \{0,1,\text{“don't know”}\}$. Call the random string r *good for input* (x,y) if $P(x,y,r) \in \{0,1\}$. That is, P computes f correctly. Since P is 0 error, $\Pr[r \text{ is good for } (x,y)] \geq 1/2$. Sample $3n$ random strings $r^{(1)},\ldots,r^{(3n)}$ independently. Consider an input $(x,y) \in \{0,1\}^n \times \{0,1\}^n$.

$$\Pr[\text{none of } r^{(1)},\ldots,r^{(3n)} \text{ is good for } (x,y)] \leq 8^{-n} \ .$$

Thus, by a union bound over $\{0,1\}^n \times \{0,1\}^n$, we obtain

$$\Pr[\exists (x,y) \in \{0,1\}^n \times \{0,1\}^n : \text{ none of } r^{(1)},\ldots,r^{(3n)} \text{ is good for } (x,y)] \leq 2^{-n} \ .$$

Thus, there exists a fixed choice $r^{(1)},\ldots,r^{(3n)}$ of strings such that for each $(x,y) \in \{0,1\}^n \times \{0,1\}^n$, at least one such string is good.

We define a new, deterministic protocol P'. In P' Alice and Bob now iterate over all these strings. For each $1 \leq i \leq 3n$, they simulate the zero-error protocol $P(x,y,r^{(i)})$. If in this simulated protocol, Bob outputs 0 or 1, then in P', Bob outputs the same. If in the simulated protocol, Bob outputs "don't know", then in P' Bob moves to its $w+1^{\text{th}}$ memory state to remember that this current simulation has failed. Since there is some $r^{(i)}$ that is good for (x,y), Alice and Bob will at some point arrive at a simulation that does not fail, but outputs the correct value. In addition to the message length required by P, the protocol P' requires additional $\lceil \log(3n) \rceil$ bits, since Alice has to tell Bob the index i of the current random string. □

6 PSPACE^{cc} and 5-state protocols

According to Theorem 5, if instead of 3 we have 5 or more states and the message length is still polylogarithmic, then we *fully* characterize PSPACE^{cc}. As mentioned before, this is the first characterization of PSPACE^{cc} in terms of "space".

Proof of Theorem 5. Let $f \in \mathsf{PSPACE}^{cc}$. By the connection between circuits and communication classes (Observation 10 below), Alice and Bob can preprocess their inputs x,y and then evaluate a circuit of $\mathsf{polylog}(n)$ depth. To be more precise, there is $m \leq 2^{\mathsf{polylog}(n)}$, functions $A,B : \{0,1\}^n \to \{0,1\}^m$ and a circuit C on $2m$ variables of depth $\mathsf{polylog}(n)$ such that $f(x,y) = C(A(x),B(y))$.

Theorem 17 (Generalized Version of Barrington's Theorem)**.** *There exists a universal constant $c \in \mathbb{N}^+$ such that if a function f can be computed by a depth d fan-in 2 circuit over basis $\{\vee,\wedge,\neg\}$, it can also be computed by a width 5 length c^d branching program.*

Thus, there is a width-5 branching program of length $2^{\mathsf{polylog}(n)}$ that computes C. Alice and Bob use A and B to preprocess their inputs into $A(x)$ and $B(y)$. Then the use Fact 8 to evaluate C with 5 memory states and $\mathsf{polylog}(n)$ message length on input $A(x),B(y)$.

For the converse, let $f \in \mathsf{SPACE}_{\mathsf{LTD}}[\mathsf{polylog}(n), 5]$. That is, f is computable by a protocol $\mathcal{P}$, and denote by $S(n) = \mathsf{polylog}(n)$ the message length requirement of $\mathcal{P}$. We will show that $f \in \mathsf{PSPACE}^{cc}$, which amounts to constructing a quantified boolean formula as in Definition 4.

Using the same notation as in Definition 6, for a fixed input (x, y), the configuration C of a particular round of $\mathcal{P}$ is denoted by $C \stackrel{\text{def}}{=} (\alpha, i, \gamma)$, where i is the round number, α is the message sent by Alice, and γ is Bob's memory state. Such a configuration is x-consistent if $P_A(i, x) = \alpha$. Two configurations $C_1 \stackrel{\text{def}}{=} (\alpha_1, i_1, \gamma_1)$ and $C_2 \stackrel{\text{def}}{=} (\alpha_2, i_2, \gamma_2)$ are adjacent if $i_2 = i_1 + 1$ and there exists $\beta \in \{0, 1, \perp\}$ such that $T_B(y, \gamma_1, \alpha_2) = (\beta, \gamma_2)$.

Therefore, to characterize the condition that $f(x, y) = 1$, we only need to certify that there is a sequence of configurations $C_0, C_1, \ldots, C_l$ such that

- $l \leq 5 \cdot 2^{S(n)}$
- for each $i \in \{1, 2, \ldots, l\}$, C_i is x-consistent
- for each $i \in \{0, 1, \ldots, l-1\}$, C_i and C_{i+1} is adjacent
- $C_0 = (\emptyset, 0, \gamma_0)$ is the designated initial configuration, γ_0 is Bob's designated initial memory state, and $\emptyset$ is a placeholer
- C_{l-1} and C_l lead Bob to output 1

We encode this computation (sequence of configurations) as a quantified boolean formula of length a logarithm of the computation size by (i) existentially guessing the middle configuration and (ii) universally asserting the correctness of both parts – i.e. exactly as in the standard theorem by Stockmeyer and Meyer [24] (see also 4.13 in [5]). The only thing to observe and conclude is that certifying correctness for Alice through ψ and for Bob through ϕ can be done completely independently which means that the unquantified formula is always in the form $\psi(u, x) \diamond \phi(u, y)$. □

7 Future Directions

We are just beginning to understand the power of the one-way limited memory model. For models that are not completely memoryless we cannot prove any lower bounds so far. Given the various connections between this model and the communication complexity polynomial hierarchy, and the fact that separating classes higher up in the communication complexity hierarchy remains an open problem (e.g. as far as we know, it is open whether $\Sigma_2^{cc} = \Pi_2^{cc}$), any new lower bound technique would bring notable advancement to the field.

We do not have lower bounds on *randomized* versions of memoryless protocols. Indeed, the most natural randomized version contains the class AM^{cc} (see Babai et al. [6] or Klauck [16]); proving lower bounds on AM^{cc} would be significant progress and is considered a hard problem.

Another direction that is worth exploring is the study of space-communication tradeoffs. That is, tradeoffs between the message length required by a protocol and the total amount of communication. More concrete: Is there a function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ such that $OS(f) \leq \mathsf{polylog}(n)$, but every protocol that uses $\mathsf{polylog}(n)$ message length has a superpolynomial total amount of communication?

Acknowledgements

We would like to thank Paul Beame, Kristoffer A. Hansen, Aram Harrow for helpful comments and suggestions on this topic.

We are also thankful to Paul Beame for communicating Shearar's proof of width-2 branching program length lower bound of "modulo 3" function.

We want to thank Prahladh Harsha and Andrej Bogdanov: The idea to look at Gap Hamming Distance came up during a discussion with them. Dominik Scheder is especially thankful to Justin Thaler, who pointed him to several related results.

References

[1] S. Aaronson and A. Wigderson. Algebrization: A new barrier in complexity theory. *TOCT*, 1(1), 2009.

[2] M. Ajtai. Σ_1^1-formulae on finite structures. *Annals of Pure and Applied Logic*, 24(1):1–48, 1983.

[3] E. Allender and U. Hertrampf. Depth reduction for circuits of unbounded fan-in. *Inf. Comput.*, 112(2):217–238, 1994.

[4] N. Alon, Y. Matias, and M. Szegedy. The space complexity of approximating the frequency moments. *Journal of Computer and System Sciences*, 58(1):137–147, 1999. Preliminary version in *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, pages 20–29, 1996.

[5] S. Arora and B. Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.

[6] L. Babai, P. Frankl, and J. Simon. Complexity classes in communication complexity theory (preliminary version). In *FOCS*, pages 337–347. IEEE Computer Society, 1986.

[7] D. A. Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in nc^1. *J. Comput. Syst. Sci.*, 38(1):150–164, 1989.

[8] E. Blais, J. Brody, and K. Matulef. Property testing lower bounds via communication complexity. In *Proceedings of the 26th Annual IEEE Conference on Computational Complexity*, pages 210–220, 2011.

[9] B. Bollobás. *Combinatorics: Set Systems, Hypergraphs, Families of Vectors, and Combinatorial Probability*. Cambridge University Press, 1986.

[10] M. Braverman. Poly-logarithmic independence fools bounded-depth boolean circuits. *Commun. ACM*, 54(4):108–115, 2011.

[11] J. Brody, S. Chen, P. A. Papakonstantinou, H. Song, and X. Sun. Space-bounded communication complexity. In R. D. Kleinberg, editor, *ITCS*, pages 159–172. ACM, 2013.

[12] R. Impagliazzo, V. Kabanets, and A. Kolokolova. An axiomatic approach to algebrization. In M. Mitzenmacher, editor, *STOC*, pages 695–704. ACM, 2009.

[13] R. Impagliazzo and R. Williams. Communication complexity with synchronized clocks. In *Proceedings of the 25th Annual IEEE Conference on Computational Complexity*, pages 259–269, 2010.

[14] P. Indyk and D. Woodruff. Tight lower bounds for the distinct elements problem. In *Proceedings of the 45th Annual Symposium on Foundations of Computer Science*, pages 283–289, 2003.

[15] M. Karchmer and A. Wigderson. Monotone circuits for connectivity require super-logarithmic depth. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing*, pages 539–550, 1988.

[16] Hartmut Klauck. On arthur merlin games in communication complexity. In *IEEE Conference on Computational Complexity*, pages 189–199. IEEE Computer Society, 2011.

[17] E. Kushilevitz and N. Nisan. *Communication Complexity*. Cambridge University Press, 1997.

[18] T. W. Lam and W. L. Ruzzo. Results on communication complexity classes. *J. Comput. Syst. Sci.*, 44(2):324–342, 1992.

[19] I. Newman. Private vs. common random bits in communication complexity. *Inf. Process. Lett.*, 39(2):67–71, 1991.

[20] R. Raz and A. Wigderson. Monotone circuits for matching require linear depth. *Journal of the ACM*, 39(3):736–744, 1992.

[21] A. A. Razborov. Lower bounds on the size of bounded depth networks over a complete basis with logical addition (russian). *Matematicheskie Zametki*, 41(4):598–607, 1.

[22] Shearer. Shearer's width 2 lower bound. from private communication with Paul Beame.

[23] R. Smolensky. Algebraic methods in the theory of lower bounds for boolean circuit complexity. In A. V. Aho, editor, *STOC*, pages 77–82. ACM, 1987.

[24] L. J. Stockmeyer and A. R. Meyer. Word problems requiring exponential time: Preliminary report. In A. V. Aho, A. Borodin, R. L. Constable, R. W. Floyd, M. A. Harrison, R. M. Karp, and H. R. Strong, editors, *STOC*, pages 1–9. ACM, 1973.

[25] H. Vollmer. *Introduction to Circuit Complexity - A Uniform Approach*. Springer, 1999.

[26] Emo Welzl. Boolean satisfiability – combinatorics and algorithms (lecture notes), 2005. `http://www.inf.ethz.ch/~emo/SmallPieces/SAT.ps`.

Appendix

A Proof of Lemma 13

Lemma 18 (Lemma 13, restated for convenience)**.** *There exists* $f, g \in \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$*, such that*

- $f \in \Sigma_2^{cc}$ *and* $g \in \Pi_2^{cc}$

- *both f and g are extensions of the partial function "Gap Hamming Distance"* GHD
- $S(f) = \Omega(n)$ *and* $S(g) = \Omega(n)$

Proof. It suffices to prove this statement for g: That is, there is a total function $g : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ that extends GHD, $g \in \Pi_2^{cc}$, and $S(g) = \Omega(n)$. We then set $f(x,y) \stackrel{\text{def}}{=} \neg g(\bar{x}, y)$. It is not difficult to see that f has the properties stated in the corollary provided g does.

We now construct the function g. We use the concept of approximate majority:

Lemma 19 (Ajtai [2])**.** *There is a polynomial-size circuit C of depth 3 whose output gate is an AND-gate; whose middle layer consists of OR-gates; and whose bottom gates are AND-gates of fan-in $O(\log(n))$. Furthermore, C computes an approximate majority. That is, $C(x) = 0$ if $|x|_1 \leq n/3$ and $C(x) = 1$ if $|x|_1 \geq 2n/3$.*

We want to emphasize that C computes a *total* function $\{0,1\}^n \to \{0,1\}$, although we know nothing about its behavior on $x \in \{0,1\}^n$ with $n/3 < |x|_1 < 2n/3$. Note that $g(x,y) \stackrel{\text{def}}{=} C(x \oplus y)$ extends Gap Hamming Distance: $g(x,y) = 0$ if $d_H(x,y) \leq n/3$ and $g(x,y) = 1$ if $d_H(x,y) \geq 2n/3$. First, we show that g has polynomial-size circuits of depth 3: The bottom gates of C are AND-gates of fan-in $O(\log n)$. In g, this gate computes an AND of $O(\log n)$ many expressions $x_i \oplus y_i$. This is a function on $O(\log n)$ variables and thus can be written as a DNF formula (depth-2 with an OR-gate at the top) of polynomial size. This DNF can be merged with the OR-gates in the middle layer of C, which gives a depth-3 circuit C' of polynomial size computing $g(x,y) = C(x \oplus y)$. Since every depth-3 circuit of polynomial size with an AND-gate at the top computes a function in Π_2^{cc}, we conclude that $g \in \Pi_2^{cc}$. This proves the first two bullet points of Corollary 13.

Lemma 20. *Let $A, B \subseteq \{0,1\}^n$ be such that $d_H(a,b) > n/3$ for all $a \in A$ and $b \in B$. Then $|A| \cdot |B| \leq n4^{H(1/3)n}$.*

With this lemma, we can now show that all monochromatic rectangles of g are very small, and thus $S(g)$ is large. Let $A \times B \subseteq \{0,1\}^n \times \{0,1\}^n$ be a 1-monochromatic rectangle. Since $g(x,y) = 0$ if $d_H(x,y) \leq n/3$, we conclude that $d_H(a,b) > n/3$ for all $(a,b) \in A \times B$. By Lemma 20, it holds that $|A| \cdot |B| \leq n4^{H(1/3)n}$. Now suppose $A \times B$ is 0-monochromatic. Since $g(x,y) = 1$ if $d_H(x,y) \leq 2n/3$, we conclude that $d_H(a,b) < 2n/3$ for all $(a,b) \in A \times B$. From a string $x \in \{0,1\}^n$ we obtain $\bar{x}$ by replacing every 0 by 1 and vice versa. We define $-B \stackrel{\text{def}}{=} \{\bar{b} \mid b \in B\}$. Note that $d_H(a,b) + d_H(a,\bar{b}) = n$ for all $a, b \in \{0,1\}^n$. Therefore $d_H(a,\bar{b}) > n/3$ for all $(a,b) \in A \times B$. This means we can apply Lemma 20 to $A \times (-B)$ and conclude that $|A| \cdot |B| = |A| \cdot |-B| \leq n4^{H(1/3)n}$.

We see that under the uniform distribution, every monochromatic rectangle has probability at most $\epsilon \stackrel{\text{def}}{=} n4^{H(1/3)n-n}$. The rectangle lower bound (Theorem 2) proves that

$$S(f) \geq \frac{1}{4}\log\left(\frac{1}{\epsilon}\right) - \frac{1}{2} \geq \frac{1}{4}\log\left(\frac{4^{n-H(1/3)n}}{n}\right) - \frac{1}{2} = \frac{1-H(1/3)}{2} \cdot n - \frac{1}{4}\log(n) - \frac{1}{2} \in \Omega\left(0.04n\right) \ .$$

This finishes the proof of Corollay 13. □

Proof of Lemma 20. Intuitively, we show that the "optimal" choice of A, B is to let A and B be Hamming balls of radius $n/3$, centered at $(0,\ldots,0)$ and $(1,\ldots,1)$, respectively. The next lemma shows that it is optimal to let A and B be Hamming balls, though it does not say anything about the optimal radii. In the following, we define $\binom{n}{\leq r} \stackrel{\text{def}}{=} \sum_{i=0}^{\lfloor r \rfloor} \binom{n}{i}$.

Lemma 21 (Far Sets)**.** *Let $A, B \subseteq \{0,1\}^n$. Let d be an integer such that $d_H(a,b) \geq d$ for all $a \in A$ and $b \in B$. Then there exists an integer $0 \leq r \leq n$ such a $|A| \leq \binom{n}{\leq r+1}$ and $|B| \leq \binom{n}{\leq n-(r+d)}$.*

We apply this lemma with $d = n/3$: There is an integer $0 \leq r \leq n$ such that $|A| \leq \binom{n}{\leq r+1}$ and $|B| \leq \binom{n}{\leq n-(r+n/3)}$. Let $H(x) \stackrel{\text{def}}{=} -x\log(x) - (1-x)\log(1-x)$ be the binary entropy function. For $0 \leq x \leq 1$, define the function $h(x)$ as follows: For $0 \leq x < 1/2$, set $h(x) \stackrel{\text{def}}{=} H(x)$; for $1/2 \leq x \leq 1$ set $h(x) \stackrel{\text{def}}{=} 1$. Note that h is continuous, monotonically increasing, and concave. It is well-known (see for example Lemma 5.6 of [26]) that $\binom{n}{\leq xn} \leq 2^{h(x)n}$. Note that $|A| \leq \binom{n}{\leq r+1} \leq n\binom{n}{\leq r}$. Write $r = \rho n$. Then

$$|A| \cdot |B| \leq n2^{h(\rho)n}2^{h(1-(\rho+1/3))n} = n2^{(h(\rho)+h(1-\rho-1/3))n} .$$

The entropy function H is concave, and so is h, being the minimum of two concave functions. We apply Jensen's inequality to obtain

$$n2^{(h(\rho)+h(1-\rho-1/3))n} \leq n2^{2h\left(\frac{\rho+1-\rho-1/3}{2}\right)n} = n4^{h\left(\frac{1-1/3}{2}\right)n} = n4^{H\left(\frac{1}{3}\right)n} .$$

The last inequality holds since $h(1/3) = H(1/3)$. This finishes the proof of Lemma 20. □

Proof of Lemma 21. For a set $A \subseteq \{0,1\}^n$, we define $A_i \stackrel{\text{def}}{=} \{x \in \{0,1\}^n \mid \exists a \in A \text{ with } d_H(x,a) \leq i\}$. For example, $A_0 = A$, and A_1 is the union of A and its neighboring vertices. In general, A_{i+1} is the union of A_i and its neighboring vertices. We use the *vertex isoperimetric inequality*:

Lemma 22 (Vertex Isoperimetric Inequality, see e.g. Theorem 5, Chapter 16, p128 in Bollobás [9])**.** *Let $n \in \mathbb{N}^+$, $A \subseteq \{0,1\}^n$, and let $0 \leq r \leq n$ be an integer. If $|A| \geq \binom{n}{\leq r}$, then $A_1 \geq \binom{n}{\leq r+1}$.*

Let r be an integer such that $\binom{n}{\leq r} \leq |A| \leq \binom{n}{\leq r+1}$. Applying the vertex isoperimetric inequality $d-1$ times to A, we conclude that $|A_{d-1}| \geq \binom{n}{\leq r+d-1}$. Note that $A_{d-1} \cap B = \emptyset$: Indeed, otherwise there is some $b \in B$ that is also in A_{d-1}. By the definition of A_{d-1}, there is some $a \in A$ with $d_H(a,b) \leq d-1$, contradicting the assumption in the theorem. Since A_{d-1} and B are disjoint, we conclude that $|A_{d-1}| + |B| \leq 2^n$. From this we obtain

$$\begin{aligned} |B| &\leq 2^n - |A_{d-1}| \leq 2^n - \binom{n}{\leq r+d-1} \\ &= 2^n - \sum_{i=0}^{r+d-1}\binom{n}{i} = \sum_{i=r+d}^{n}\binom{n}{i} = \sum_{j=0}^{n-(r+d)}\binom{n}{j} = \binom{n}{\leq n-(r+d)} . \end{aligned}$$

We conclude that $|A| \leq \binom{n}{\leq r+1}$ and $|B| \leq \binom{n}{\leq n-(r+k)}$, as stated. □

B Proof of Theorem 9

Suppose $f(x,y)$ can be decomposed as $g(h_1(x,y), \ldots, h_k(x,y))$ where each h_i depends on at most ℓ variables. We give a protocol where Bob has two memory states and message length $k + \ell + \lceil \log k \rceil$.

The protocol proceeds in phases. In each phase, Alice and Bob guess some $z \in \{0,1\}^k$ and want to verify that $h_i(x,y) = z_i$ for all $1 \leq i \leq k$. To do so, they use k rounds. In round i, they evaluate $h_i(x,y)$. Note that h_i depends on at most ℓ variables. Some of them belong to Alice. Alice sends

those bits, and Bob can evaluate $h_i(x, y)$. If $h_i(x, y) \neq z_i$, he moves into his second memory state, remembering that z was a wrong guess. Once i reaches k, Bob remembers whether the guess z was correct or not. If is was not, he continues, and they move on to the next guess $z \in \{0,1\}^k$. Otherwise, he knows that $z_i = h_i(x, y)$ for all $1 \leq i \leq k$, and outputs $g(z_1, \ldots, z_k)$.

In each step, Alice sends Bob up to ℓ bits of her own input. Additionally, she has to send Bob the "guess" $z \in \{0,1\}^k$, and finally the counter $i \in \{1, \ldots, k\}$. Thus, her message consists of at most $\ell + k + \lceil \log k \rceil$ bits.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de